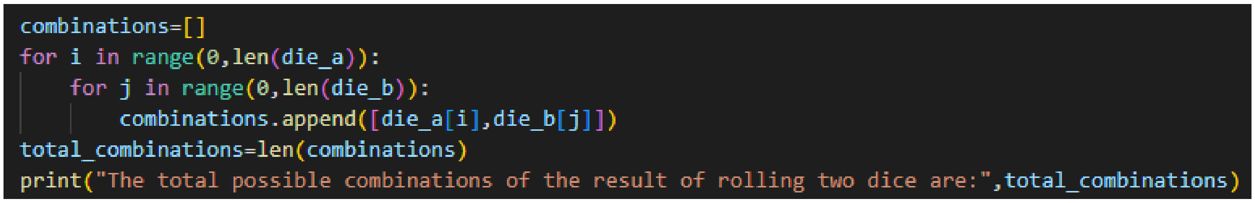
DOOMED DICE

PART 1

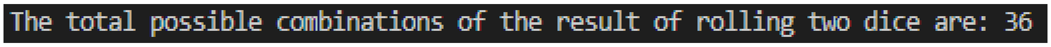
1. When one die is rolled, there are six possible outcomes. When a second die is rolled with it, there are 6*6=**36 different possible outcomes**, since each side of the first die has a chance of coming up with each of the six faces of the other die.

   Code :



```
combinations=[]
for i in range(0,len(die_a)):
    for j in range(0,len(die_b)):
        combinations.append([die_a[i],die_b[j]])
total_combinations=len(combinations)
print("The total possible combinations of the result of rolling two dice are:",total_combinations)
```

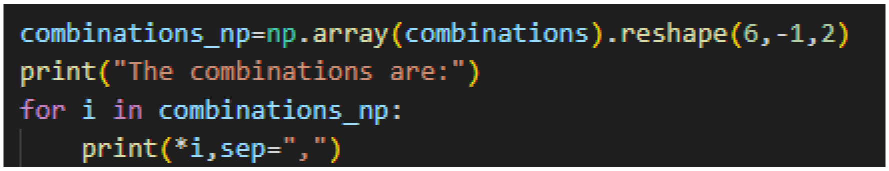
   Output :



```
The total possible combinations of the result of rolling two dice are: 36
```

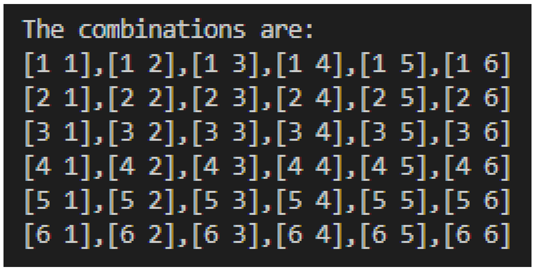
2. The possible combinations are identified in the previous operation, and are reshaped to be better represented

   Code :



```
combinations_np=np.array(combinations).reshape(6,-1,2)
print("The combinations are:")
for i in combinations_np:
    print(*i,sep=",")
```

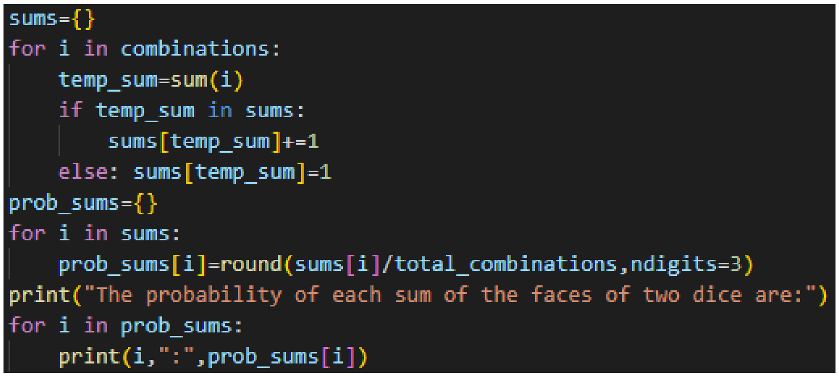
   Output:



```
The combinations are:
[1 1],[1 2],[1 3],[1 4],[1 5],[1 6]
[2 1],[2 2],[2 3],[2 4],[2 5],[2 6]
[3 1],[3 2],[3 3],[3 4],[3 5],[3 6]
[4 1],[4 2],[4 3],[4 4],[4 5],[4 6]
[5 1],[5 2],[5 3],[5 4],[5 5],[5 6]
[6 1],[6 2],[6 3],[6 4],[6 5],[6 6]
```

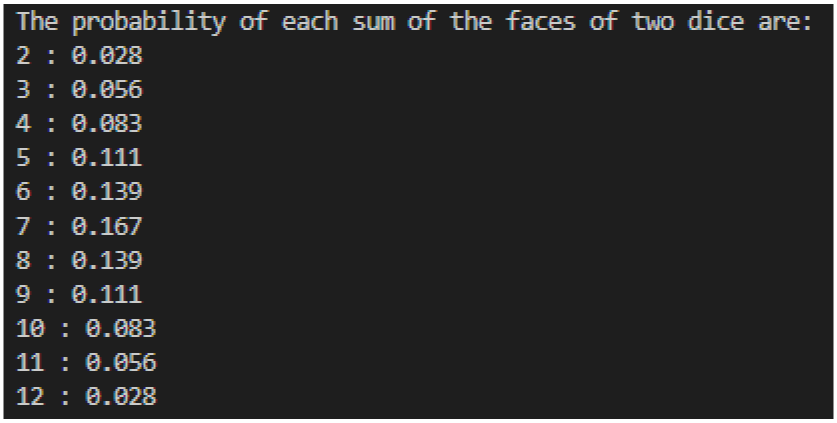
3. The probability of each sum can be calculated and stored in a dictionary in key-value pairs. The probability of a sum appearing is calculated as
   P(sum=x) = (No. of times X shows up as sum)/(All possible outcomes)

   Code :



```
sums={}
for i in combinations:
    temp_sum=sum(i)
    if temp_sum in sums:
        sums[temp_sum]+=1
    else: sums[temp_sum]=1
prob_sums={}
for i in sums:
    prob_sums[i]=round(sums[i]/total_combinations,ndigits=3)
print("The probability of each sum of the faces of two dice are:")
for i in prob_sums:
    print(i,":",prob_sums[i])
```

Output :



```
The probability of each sum of the faces of two dice are:
2 : 0.028
3 : 0.056
4 : 0.083
5 : 0.111
6 : 0.139
7 : 0.167
8 : 0.139
9 : 0.111
10 : 0.083
11 : 0.056
12 : 0.028
```

PART 2

The dice are cleared, and are supposed to be reset with new values. The dice may now have non-unique faces, and the constraints are
A. no face on Die A may be greater than 4
B. The probabilities of the sums of outcomes must be unchanged
The sums produced by rolling two dice are represented by this table:

Faces	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

Since Die A cannot have any faces greater than 4, the last two columns are unattainable and need to be replaced.
Now, we know that P(sum=2)=1/36, from the original distribution. To maintain this distribution while retaining the original probabilities, we deduce that exactly one face on each die must be 1 Additionally, since the largest face possible on Die A is 4, to maintain P(sum=12)=1/36, exactly one face on Die A must be 4, and correspondingly, exactly one face on die B must be 8.

This means Die A ranges from [1,1,1,1,1,4] to [1,3,3,3,3,4], excluding all possibilities where die_a[x]>4, and Die B ranges from [1,1,1,1,1,8] to [1,7,7,7,7,8].

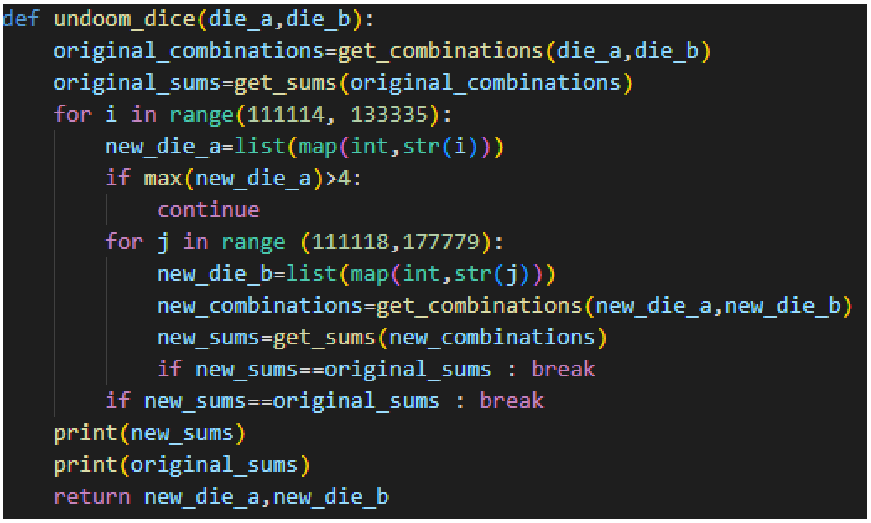
We iterate our method that finds sum distributions over all these possibilities, which returns one possible combination

The final sum distribution for the new pair of Dice is represented in the following table

Faces	1	2	2	3	3	4
1	2	3	3	4	4	5
3	4	5	5	6	6	7
4	5	6	6	7	7	8
5	6	7	7	8	8	9
6	7	8	8	9	9	10
8	9	10	10	11	11	12

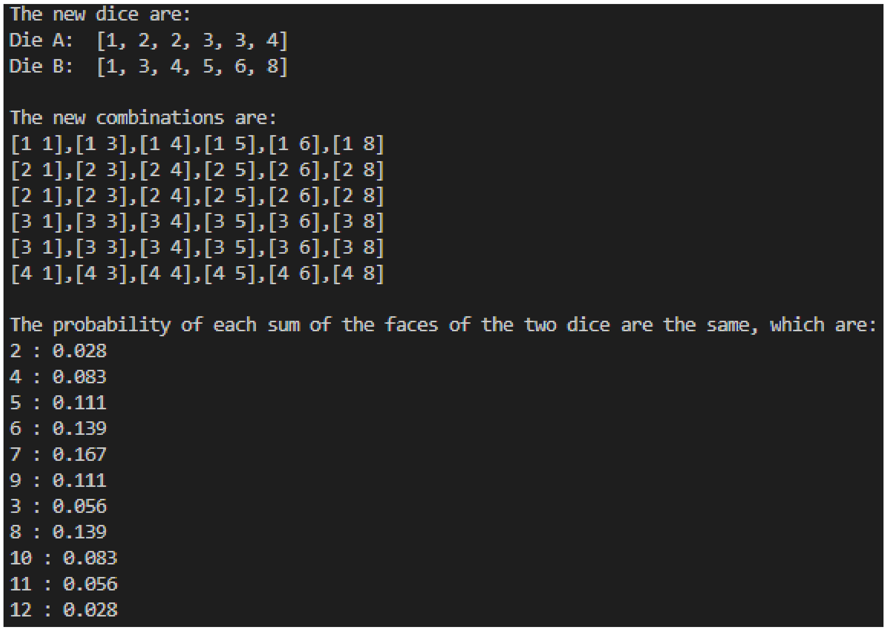
The new Dice are:
Dice A : [1,2,2,3,3,4]
Dice B : [1,3,4,5,6,8]

Code :



```python
def undoom_dice(die_a,die_b):
    original_combinations=get_combinations(die_a,die_b)
    original_sums=get_sums(original_combinations)
    for i in range(111114, 133335):
        new_die_a=list(map(int,str(i)))
        if max(new_die_a)>4:
            continue
        for j in range (111118,177779):
            new_die_b=list(map(int,str(j)))
            new_combinations=get_combinations(new_die_a,new_die_b)
            new_sums=get_sums(new_combinations)
            if new_sums==original_sums : break
        if new_sums==original_sums : break
    print(new_sums)
    print(original_sums)
    return new_die_a,new_die_b
```

Output:



```
The new dice are:
Die A:  [1, 2, 2, 3, 3, 4]
Die B:  [1, 3, 4, 5, 6, 8]

The new combinations are:
[1 1],[1 3],[1 4],[1 5],[1 6],[1 8]
[2 1],[2 3],[2 4],[2 5],[2 6],[2 8]
[2 1],[2 3],[2 4],[2 5],[2 6],[2 8]
[3 1],[3 3],[3 4],[3 5],[3 6],[3 8]
[3 1],[3 3],[3 4],[3 5],[3 6],[3 8]
[4 1],[4 3],[4 4],[4 5],[4 6],[4 8]

The probability of each sum of the faces of the two dice are the same, which are:
2 : 0.028
4 : 0.083
5 : 0.111
6 : 0.139
7 : 0.167
9 : 0.111
3 : 0.056
8 : 0.139
10 : 0.083
11 : 0.056
12 : 0.028
```